10-20-00

A

10/19/00 Jc946 U.S. PTO

ATTORNEY DOCKET NO.
016295.0620
(DC-02500)

PATENT APPLICATION

Express Mail # EL609945211US
Date of Mailing: October 19, 2000

APPLICATION FOR U.S. PATENT UNDER 37 C.F.R. 1.53(b)
TRANSMITTAL FORM

Jc542 U.S. PTO 09/692647 10/19/00

Box Patent Application
ASSISTANT COMMISSIONER FOR PATENTS
Washington, D.C. 20231

Sir:

Transmitted herewith for filing is the patent application of:

Inventor or Application Identifier: Martin McAfee

Entitled: System and Method for Debugging Multiprocessor Systems

Enclosed are:

Enclosed are:	X	Specification(10 pages)
	X	Claims (6 pages)
	X	Abstract (1 page)
	X	Drawing(s) (2 Sheets Formal)

X Signed combined Declaration and Power of Attorney.

X Return Receipt Postcard

X An Assignment of the invention to **DELL PRODUCTS L.P.** is attached. A cover sheet in compliance with 37 C.F.R. §§ 3.28 and 3.31 is included with the Assignment recordation fee of $40.00 pursuant to 37 C.F.R. § 1.21(h).

Applicant is large Entity.

HOU03:726228.1

Attorney's Docket:
016295.0612
(DC-02195)
Page 2

PATENT APPLICATION

FEE CALCULATION					FEE
	Number		Number Extra	Rate	Basic Fee $ 710.00
Total Claims:	22	-20 =	2	X $18 =	$36.00
Independent Claims	3	-3 =	0	X $78 =	$0.00
				TOTAL FILING FEE =	$746.00

Enclosed is a check in the amount of $746.00 to satisfy filing fee requirements under 37 C.F.R. § 1.16. Please charge any additional fees or credit any overpayment to Deposit Account No. 02-0383 of BAKER & BOTTS, L.L.P. **A duplicate copy of this sheet is enclosed.**

Respectfully submitted,

BAKER BOTTS, L.L.P.
Attorneys for Applicants

Melissa J. Szanto

MELISSA J. SZANTO
Reg. No. 40,834

Date: October 19, 2000

Correspondence Address:
Melissa J. Szanto
Baker Botts, L.L.P.
One Shell Plaza
910 Louisiana
Houston, Texas 77002-4995
Tel. (713) 229-1739
Fax. (713) 229-1522

Express Mail #EL609945211US

SYSTEM AND METHOD FOR DEBUGGING MULTIPROCESSOR SYSTEMS

Inventors: Martin McAfee
1909 Patriot Drive
Lago Vista, Texas 78645

Assignee: Dell Products L.P.
One Dell Way
Round Rock, Texas 78682-2244

BAKER BOTTS L.L.P.
One Shell Plaza
910 Louisiana
Houston, Texas 77002-4995

SYSTEM AND METHOD FOR DEBUGGING MULTIPROCESSOR SYSTEMS

TECHNICAL FIELD

The system and method described herein relates generally to the field of computer systems, and more particularly, to debugging a computer system having multiple processors.

BACKGROUND

A personal computer system typically includes a system unit having a central processing unit (CPU) and associated memory, and a storage device such as a hard disk, floppy disk or CD ROM. Also included are an input device, such as a keyboard or mouse, a display device, and possibly other peripherals as well. Many computer systems today include multiple CPUs mounted on what is commonly referred to as a motherboard, wherein each of these CPUs contain code known as the system BIOS ("Basic Input/Output System"). The system BIOS is responsible for testing system hardware and starting the operating system during the booting process. The BIOS also contains data and instructions that enables the transfer of data to and from the system hardware. The system BIOS is stored in system memory, typically in non-volatile memory such as read-only memory (ROM) or flash memory.

Before computer systems are sold, the BIOS must be "debugged," or tested to ensure that it is working properly, and if it is not then to ensure that any existing errors are detected and corrected. The motherboard of the computer typically includes a debug port provided for this purpose. A computer system having debugging software is connected to the motherboard via the debug port. The debug port is connected to each of the CPU sockets serially so that

signals and data flow serially from the debugging computer through the debug port, and then through each successive CPU before returning to the debugging computer through the debug port.

Problems arise during debugging when one of the positions or slots on a motherboard designed to carry multiple CPUs is not occupied by a CPU. In some multiprocessor systems, an empty CPU slot must always be occupied to ensure proper termination of the line or bus during normal operation of the computer. Separate terminator cards are inserted into the unoccupied CPU slots for this purpose. These terminator cards perform required termination functions, and also, during debugging procedures, function to ensure that the signal is passed through to a subsequent CPU so as to properly complete the serial debugging circuit. An unoccupied CPU slot will otherwise cause a break in the debugging circuit, preventing altogether debugging of any CPU. Insertion of separate terminator cards in unoccupied slots is disadvantageous in that these separate terminator cards are expensive, and must be manually inserted during manufacturing.

Some more recent multiprocessor systems do not need separate terminator cards to perform termination. These systems have termination capabilities built into the CPU that ensure termination. For purposes of debugging, however, empty CPU slots still must be bypassed to complete the debugging circuit path. For these types of systems, the only known method by which to bypass unpopulated CPU slots for debugging purposes is to insert a jumper between successive CPUs, and to manually adjust the jumper when undergoing debugging versus normal operation. Jumpers that require physical manipulation prior to debugging are both labor intensive and time consuming. Further, because of space

constraints it is becoming increasingly difficult to fit such jumpers between or in close proximity to the CPUs, as is required.

Accordingly, what is needed is an improved system and method for debugging multiprocessor systems that may have unoccupied CPU slots.

SUMMARY OF THE PREFERRED EMBODIMENTS

In accordance with the present disclosure, a debugging circuit capable of debugging a plurality of possible microprocessors is provided that includes a debug port, a plurality of microprocessor sockets, each of which are adapted to receive a microprocessor, and a plurality of switches, each of which correspond to a respective one of the plurality of microprocessor sockets. The plurality of microprocessor sockets are adapted to form a serial signal path, and each of the switches is capable of automatically detecting whether a microprocessor is present in the corresponding microprocessor socket. If a microprocessor is present in the corresponding microprocessor socket, then said switch is automatically configured to include the microprocessor within the signal path, and if a microprocessor is not present in the corresponding microprocessor socket then the switch is automatically configured so that the signal path bypasses the corresponding microprocessor socket.

According to one embodiment, a debugging input is provided to each microprocessor socket, and a debugging output is provided from each microprocessor that is present in the corresponding microprocessor socket. In yet another embodiment, each switch receives as an input a microprocessor detection signal indicating whether the corresponding microprocessor is present.

According to yet another embodiment, for each switch, if the microprocessor is present then the switch provides as an output the debugging output of the corresponding microprocessor. If the microprocessor is not present, then the switch provides as a switch output the debugging input to the corresponding microprocessor.

In yet another embodiment, for each switch not corresponding to a last microprocessor in the serial signal path, the switch output is provided as a debugging input to a subsequent microprocessor in the serial signal path. For the switch corresponding to the last microprocessor in the serial signal path, the switch output is provided to the debug port.

According to another embodiment, the debug port is electrically coupled to a computer and receives input from and provides output to said computer.

According to yet alternate embodiments, the plurality of switches each comprise a pair of bipolar transistors or field effect transistors.

A debugging switch is also provided for use in a debugging circuit capable of debugging a plurality of possible processors. The debugging switch includes a first node for receiving a processor detection signal indicating whether a first processor is present in a corresponding processor socket, a second node for receiving a debugging input signal to the first processor, a third node for receiving a debugging output signal from the first processor if the first processor is present in the corresponding processor socket, a fourth node for providing a switch output signal, and a switching element. If the processor detection signal indicates that the corresponding processor is not present, then the switching element is automatically configured so that the switch provides as a switch output the debugging input signal, and if the processor detection signal indicates that the corresponding processor is

present, then the switching element is automatically configured so that the switch provides as a switch output the debugging output signal.

According to one embodiment, the switching element further comprises first and second bipolar transistors, and according to an alternate embodiment the switching element comprises field effect transistors. The field effect transistors may be junction field effect transistors or insulated gate field effect transistors.

According to yet another embodiment, the switch output is provided as a debugging input to a second processor.

A method for debugging at least one of a plurality of possible microprocessors is also provided including the step of providing a debugging circuit having a plurality of microprocessor sockets adapted to form a serial signal path. Each microprocessor socket corresponds to a different one of the plurality of possible microprocessors and is capable of receiving a microprocessor. The method further includes the steps of providing a switch corresponding to each of the microprocessor sockets, providing as an input to each of the switches a processor detection signal indicating whether a microprocessor is present in the corresponding microprocessor socket, providing as an input to each of the switches a processor debugging input for the corresponding microprocessor, and providing as an input to each of the switches a processor debugging output from the corresponding processor if the microprocessor is present in the corresponding microprocessor socket. The method further includes the step of the switch providing as a switch output the processor debugging input if the corresponding microprocessor is not present in the corresponding microprocessor socket,

and providing as a switch output the processor debugging output if the microprocessor is present in the corresponding microprocessor socket.

According to one embodiment, the method further includes the step of, for each switch corresponding to a microprocessor that is not a last microprocessor in the serial signal path, providing the switch output as a debugging input to a subsequent microprocessor in the serial signal path. In yet another embodiment, the method further includes the step of, for the switch corresponding to the last microprocessor in the serial signal path, providing the switch output to a debug port.

According to yet another embodiment, the method further includes the step of providing as a debugging input to a first microprocessor in the serial signal path a signal received from the debug port.

BRIEF DESCRIPTION OF THE DRAWINGS

A more complete understanding of the disclosed embodiments and advantages thereof may be acquired by referring to the following description taken in conjunction with the accompanying drawings, in which like reference numbers indicate like features, and wherein:

FIGURE 1 illustrates a debugging circuit capable of debugging a plurality of possible microprocessors; and

FIGURE 2 illustrates one embodiment of a switch for use in a debugging circuit capable of debugging a plurality of possible microprocessors.

DETAILED DESCRIPTION

Figure 1 illustrates a motherboard 100 of a computer system including multiple microprocessors (CPUs) 102a, 102b, 102c, and 102d, and a debug port 104. It is well known

that each of the microprocessors are received by a corresponding microprocessor "socket" located on the motherboard. The microprocessor sockets are not shown in Figure 1, as each such socket is occupied by a microprocessor. The debug port is a connector that enables an external computer system 106, when coupled with the debug port via cable 105 or the like, to utilize a debugging software program to communicate with the CPUs so as to monitor and/or control execution of BIOS code within each processor. A debugging circuit 110 passes signals from debug port 104 serially to each successive microprocessor socket before returning to the debug port. As will be described more fully below, the present invention automatically ensures that the serial debugging signal path is completed regardless of whether each of the possible CPUs are present in their corresponding microprocessor sockets.

Switches 108a, 108b, 108c, 108d each correspond to a respective one of the microprocessor sockets, and each preferably is in parallel with its corresponding microprocessor socket. The switches each receive as an input a processor detection signal 110a, 110b, 110c, 110d that indicates whether a microprocessor is present in the corresponding microprocessor socket. If present, the switch will automatically be configured to include that microprocessor within the debugging signal path, and if absent, the switch will automatically be configured so that the debugging signal path bypasses the unoccupied microprocessor socket.

For example, in Figure 1, if the microprocessor detection signal 110c input to switch 108c indicates that CPU2 is not present, switch 108c will be automatically configured to provide as a debugging input to CPU3, the signal 116b that would have otherwise have been provided as the debugging input to CPU2. Thus, switch 108c automatically ensures that the

debugging signal path is complete by bypassing the unoccupied microprocessor socket. If, however, CPU2 is present, switch 108c will be automatically configured to provide as a debugging input to CPU3 the debugging output signal from CPU2, thereby ensuring that CPU2 is included within the debugging signal path.

Figure 2 illustrates one embodiment of such a switch in greater detail. For purposes of illustration, switch 108c is shown, although it will be understood that the illustrated circuit may be any of the switches shown in Figure 1. The debugging circuit receives a microprocessor detection signal 110c that indicates whether CPU2 is present at node 206, which is electrically coupled to the microprocessor socket that is capable of receiving CPU2. According to one embodiment in which the microprocessors are Intel processors, the CPU detection signal is a SKTOCC signal, which asserts a positive voltage on node 206 when CPU2 is not present, and is grounded when CPU2 is present. Switch 108c also includes another node 204 for receiving the microprocessor debugging output of CPU2 (112c), if the CPU is present in its corresponding socket. Switch 108c includes yet another node 202 for receiving the microprocessor debugging input to CPU2 (116b). For the Intel processors referred to above, the CPU2_TDI signal is the debugging input signal for CPU2 and CPU2_TDO is the debugging output signal for CPU2. Finally, the output of switch 108c provides the debugging input signal to CPU3 (116c) (i.e., CPU3_TDI) via node 212.

The embodiment of Figure 2 further illustrates a typical pair of bipolar transistors arranged in a well known totem pole configuration so that, depending on the input at node 206, one transistor will be turned on while the other is turned off. Thus, the signal path for the debugging circuit will either be from node 202 to node 212 so that the input to CPU2

(116b) will be directly provided as the input to CPU3, or from node 204 to node 212, so that output of CPU2 (112c) will be provided as the input to CPU3. In other words, the input (116b) to CPU2 will directly become the input to CPU3 (116c) if CPU2 is not present, thereby effectively bypassing CPU2. On the other hand, if CPU2 is present, the output to CPU2 (112c) will become the input to CPU3 (116c), ensuring that CPU2 is included within the signal path if present.

More specifically, for transistor 208, the application of a positive voltage (i.e., 3.3V) at node 206, which in the Intel embodiment described indicates that CPU2 is not present, will cause a positive voltage to be applied to the base of the transistor, which in turn will cause the current to flow from node 202 through the transistor to node 212. Thus, if CPU2 is not present, switch 108c transfers the input to CPU2 directly to the input to CPU3, bypassing the unoccupied CPU2 slot. For transistor 210, the application of a positive voltage at node 206 turns off the transistor preventing current flow therethrough. If node 206 is grounded, however, such as is the case when CPU2 is present, then the application of this voltage level to the base of transistor 210 turns on the transistor, enabling current flow from node 204 to output node 212. When applied to the base of transistor 208, transistor 208 will be shut off, preventing current flow therethrough. Thus, when CPU2 is present, switch 108c conveys the debugging output from CPU2 directly as the debugging input for CPU3.

Resistors 214 and 216 shown in Figure 2 are a well known pull up/pull down resistor pair used to ensure that the signal at node 212 is strong enough to maintain the integrity of the signal path. According to one embodiment, the voltage V+ applied at pull up resistor 214 is the CPU core voltage, which is typically anywhere below 2.2 volts. It will be understood by

those skilled in the art that the positive voltage applied at node 206, such as to indicate the absence of CPU2 as described above, must be sufficiently above the CPU core voltage to properly enable current flow through the transistors. Therefore, this voltage should be high enough to accommodate varying CPU core voltages.

Further, although the embodiment of Figure 2 illustrates the use of bipolar type transistors, one skilled in the art will readily recognize that the circuit of switches 108 may be constructed using any type of field effect transistors, both junction and insulated gate.

Although the present invention has been described in detail, it should be understood that various changes, substitutions and alterations can be made hereto without departing from the spirit and scope of the invention as defined by the appended claims.

WHAT IS CLAIMED IS:

1. A debugging circuit capable of debugging a plurality of possible microprocessors, comprising:

a debug port;

a plurality of microprocessor sockets, each of said microprocessor sockets adapted to receive a microprocessor;

a plurality of switches, each of said plurality of switches corresponding to a respective one of said plurality of microprocessor sockets;

wherein said plurality of microprocessor sockets are adapted to form a serial signal path, and wherein each of said switches is capable of automatically detecting whether a microprocessor is present in the corresponding microprocessor socket, and if a microprocessor is present in said corresponding microprocessor socket then said switch is automatically configured to include said microprocessor within said signal path, and if a microprocessor is not present in said corresponding microprocessor socket then said switch is automatically configured so that said signal path bypasses said corresponding microprocessor socket.

2. The debugging circuit according to claim 1, wherein a debugging input is provided to each microprocessor socket, and wherein a debugging output is provided from each microprocessor that is present in said corresponding microprocessor socket.

3. The debugging circuit according to claim 2, wherein each switch receives as an input a microprocessor detection signal indicating whether said corresponding microprocessor is present.

4. The debugging circuit according to claim 3, wherein for each switch, if said microprocessor is present then said switch provides as an output said debugging output of said corresponding microprocessor.

5. The debugging circuit according to claim 4, wherein for each switch, if said microprocessor is not present, then said switch provides as a switch output said debugging input to said corresponding microprocessor.

6. The debugging circuit according to claim 5, wherein for each switch not corresponding to a last microprocessor in said serial signal path, said switch output is provided as a debugging input to a subsequent microprocessor in said serial signal path.

7. The debugging circuit according to claim 6, wherein for said switch corresponding to said last microprocessor in said serial signal path, said switch output is provided to said debug port.

8. The debugging circuit according to claim 7, wherein said debug port is electrically coupled to a computer and receives input from and provides output to said computer.

9. The debugging circuit according to claim 8, wherein said plurality of switches each comprise a pair of bipolar transistors.

10. The debugging circuit according to claim 8, wherein said plurality of switches each comprise field effect transistors.

11. A debugging switch for use in a debugging circuit capable of debugging a plurality of possible processors comprising:

a first node for receiving a processor detection signal indicating whether a first processor is present in a corresponding processor socket;

a second node for receiving a debugging input signal to said first processor;

a third node for receiving a debugging output signal from said first processor if said first processor is present in said corresponding processor socket;

a fourth node for providing a switch output signal; and

a switching element, wherein if said processor detection signal indicates that said corresponding processor is not present, then said switching element is automatically configured so that said switch provides as a switch output said debugging input signal, and wherein if said processor detection signal indicates that said corresponding processor is

present, then said switching element is automatically configured so that said switch provides as a switch output said debugging output signal.

12. The debugging switch according to claim 11, wherein said switching element further comprises first and second bipolar transistors.

13. The debugging switch according to claim 12, wherein said switching element comprises field effect transistors.

14. The debugging switch according to claim 13, wherein said field effect transistors are junction field effect transistors.

15. The debugging switch according to claim 13, wherein said field effect transistors are insulated gate field effect transistors.

16. The debugging switch according to claim 11, wherein said switch output is provided as a debugging input to a second processor.

17. A method for debugging at least one of a plurality of possible microprocessors, comprising the steps of:

providing a debugging circuit having a plurality of microprocessor sockets adapted to form a serial signal path, wherein each microprocessor socket corresponds to a different one of said plurality of possible microprocessors and is capable of receiving a microprocessor;

providing a switch corresponding to each of said microprocessor sockets;

providing as an input to each of said switches a processor detection signal indicating whether a microprocessor is present in said corresponding microprocessor socket;

providing as an input to each of said switches a processor debugging input for said corresponding microprocessor;

providing as an input to each of said switches a processor debugging output from said corresponding processor if said microprocessor is present in said corresponding microprocessor socket;

said switch providing as a switch output said processor debugging input if said corresponding microprocessor is not present in said corresponding microprocessor socket, and providing as a switch output said processor debugging output if said microprocessor is present in said corresponding microprocessor socket.

18. The method according to claim 17, further comprising the step of:

for each switch corresponding to a microprocessor that is not a last microprocessor in said serial signal path, providing said switch output as a debugging input to a subsequent microprocessor in said serial signal path.

19. The method according to claim 18, further comprising the step of:

for said switch corresponding to said last microprocessor in said serial signal path, providing said switch output to a debug port.

20. The method according to claim 19, further comprising the step of:

providing as a debugging input to a first microprocessor in said serial signal path a signal received from said debug port.

21. The method according to claim 20, wherein said switches each comprise a pair of bipolar transistors.

22. The method according to claim 20, wherein said switches each comprise field effect transistors.

ATTORNEY'S DOCKET
016295.0620
(DC-02500)

PATENT APPLICATION

SYSTEM AND METHOD FOR DEBUGGING MULTIPROCESSOR SYSTEMS

ABSTRACT OF THE DISCLOSURE

A debugging circuit capable of debugging a plurality of possible microprocessors, and a switch for use in the same. The debugging circuit includes a debugging port, a plurality of microprocessor sockets each adapted to receive a microprocessor, and a plurality of switches corresponding to a respective microprocessor socket. The plurality of microprocessor sockets are adapted to form a serial signal path, and each of the switches is capable of automatically detecting whether a microprocessor is present in the corresponding microprocessor socket. If a microprocessor is present, the switch is automatically configured to include the microprocessor within the signal path, and if the microprocessor is not present, the switch is automatically configured so that the signal path bypasses that microprocessor socket.

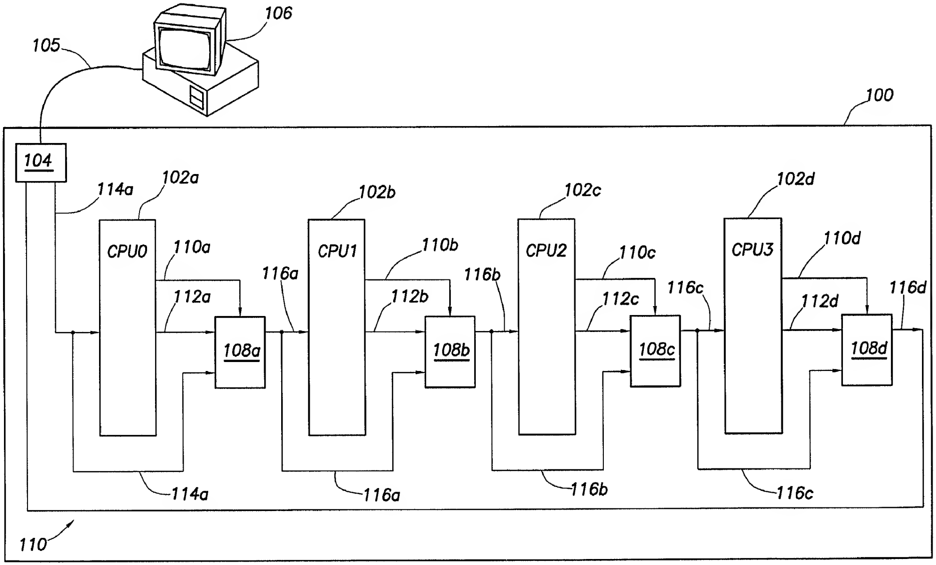
106
105
100
104
114a
102a
102b
102c
102d
CPU0
CPU1
CPU2
CPU3
110a
110b
110c
110d
112a
112b
112c
112d
116a
116b
116c
116d
108a
108b
108c
108d
114a
116a
116b
116c
110

FIG. 1

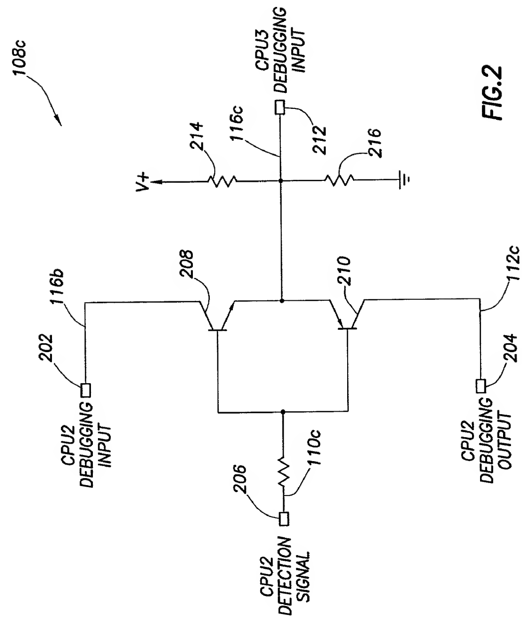
108c
CPU3 DEBUGGING INPUT
212
116c
214
216
V+
208
210
116b
112c
202
204
CPU2 DEBUGGING INPUT
CPU2 DEBUGGING OUTPUT
110c
206
CPU2 DETECTION SIGNAL

FIG.2

DECLARATION AND POWER OF ATTORNEY

As a below named inventor, I declare that:

My residence, post office address and citizenship are as stated below next to my name; that I believe I am an original, first and joint inventor of the subject matter which is claimed and for which a patent is sought on the invention or design entitled *System And Method For Debugging MultiProcessor Systems*, the specification of which (check one):

__X__ is attached hereto; or

_____ was filed on ____________________ as Application Serial No. ________________

and was amended on __________ (if applicable);

that I have reviewed and understand the contents of the above-identified specification, including the claims, as amended by any amendment referred to above; and that I acknowledge the duty to disclose to the U.S. Patent and Trademark Office all information known to me to be material to patentability as defined in 37 C.F.R. § 1.56.

I hereby claim foreign priority benefits under 35 U.S.C. § 119 of any foreign application(s) for patent or inventor's certificate listed below and have also identified below any foreign application(s) for patent or inventor's certificate having a filing date before that of the application on which priority is claimed:

Number	Country	Date Filed	Priority Claimed (Yes) (No)
NONE			

I hereby claim the benefit under 35 U.S.C. § 120 of any United States application(s) listed below and, insofar as the subject matter of each of the claims of this application is

not disclosed in the prior United States application(s) in the manner provided by the first paragraph of 35 U.S.C. § 112, I acknowledge the duty to disclose to the U.S. Patent and Trademark Office all information known to me to be material to patentability as defined in 37 C.F.R. § 1.56 which became available between the filing date of the prior application(s) and the national or PCT international filing date of this application:

Application Serial Number	Date Filed	Status
NONE		

I hereby appoint:

Henry Garrana	Reg. No. 27,887
Mark P. Kahler	Reg. No. 29,178
Anthony E. Peterman	Reg. No. 38,270
Diana L. Roberts	Reg. No. 36,654

of Dell Computer Corporation, and the following:

Scott F. Partridge	Reg. No. 28,142
Jerry W. Mills	Reg. No. 23,005
Mitchell D. Lukin	Reg. No. 30,772
Michael E. Wilson	Reg. No. 32,835
G. Alan Witte	Reg. No. 36,061
Lori D. Stiffler	Reg. No. 36,939
Ann C. Livingston	Reg. No. 32,479
Thomas R. Felger	Reg. No. 28,842
Roger J. Fulghum	Reg. No. 39,678
Robert W. Holland	Reg. No. 40,020
Claude E. Cooke, Jr.	Reg. No. 34,142
Melissa J. Szanto	Reg. No. 40,834
Michael A. Hawes	Reg. No. 38,487

all of the firm of Baker Botts L.L.P., my attorneys with full power of substitution and revocation, to prosecute this application and to transact all business in the United States Patent and Trademark Office connected therewith, and to file and prosecute any international patent applications filed thereon before any international authorities.

Send Correspondence To:
Baker Botts L.L.P.
One Shell Plaza
910 Louisiana
Houston, Texas 77002-4995

Direct Telephone Calls To:
Melissa Szanto
at (713) 229-1739
Atty. Docket No. 016295.0620

I declare that all statements made herein of my own knowledge are true and that all statements made on information and belief are believed to be true; and further that these statements were made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under Section 1001 of Title 18 of the United States Code, and that such willful false statements may jeopardize the validity of the application or any patent issuing thereon.

Full name of the inventor	Martin McAfee
Inventor's Signature	
Date	10/18/2000
Residence (City, County, State)	Lago Vista, Travis County, Texas
Citizenship	United States of America
Post Office Address	Lago Vista, Texas 78645